

HYBRIDES REQUIREMENTS ENGINEERING

Herausforderungen agiler Projekte in klassischen Organisationsstrukturen

Hybride Prozessmodelle bieten Unternehmen eine Möglichkeit, agile Projektmanagement-Ansätze in sonst eher klassischen Kontexten zu pilotieren. Oft entscheiden organisatorische Prozesse über den Erfolg oder Misserfolg eines solchen Projekts. Dieser Artikel gibt praktische Tipps für den Umgang mit den Herausforderungen der veränderten Arbeitsumgebung eines Requirements Engineers in einem hybriden Projektmodell.

rausforderungen der veränderten Arbeitsumgebung eines Requirements Engineers in einem hybriden Projektmodell.

Konsistenz über Tool-Brüche hinweg

Die Spezifikation eines agilen Projekts in einer klassischen Organisationsstruktur wird an zwei Stellen benötigt: Während der Projektlaufzeit arbeiten das Entwicklungsteam und weitere Projektbeteiligte mit einem sogenannten User Story Mapping Tool. Nach Projektabschluss nutzt

TIPPS ZUM HANDLING DES USER STORY MAPPING TOOLS.

Oft dauert es seine Zeit, um mit einer neuen Software warm zu werden. Die nachfolgenden Tipps können die Arbeit mit einem User Story Mapping Tool erleichtern.

- 1 **Epics aus Benutzersicht schreiben:** Eine einfache, nicht technische Sprache sorgt dafür, dass selbst an den Sprint Review-Meetings teilnehmende Kunden die Aufgabe verstehen.
- 2 **Epics clever strukturieren:** Um die Übersicht zu wahren, sollte ein Projekt nicht mehr als 7-10 Epics umfassen. Eines dieser Epics sollte dabei für Analyseaufgaben genutzt werden, um bspw. während des Sprints Workshops durchzuführen, sich mit anderen Abteilungen abzustimmen oder generell Lösungen zu besprechen. Diese Tätigkeiten kosten Zeit, die wie jeder andere Aufwand eingeplant werden muss. Darüber hinaus lohnt es sich, Muster-Stories und -Aufgaben anzulegen und diese immer wieder zu verwenden: Ein einheitlicher Aufbau der Stories und Aufgaben kann das Lesen erleichtern.
- 3 **Im letzten Sprint keine Kernfunktionalitäten mehr umsetzen:** Der letzte Sprint sollte dazu dienen, das Projekt sauber abzuschließen. Die Zeit kann z. B. dafür genutzt werden, noch offene Punkte aus dem vorangegangenen Sprint zu erledigen, den Code und die Spezifikation „aufzuräumen“, oder auch, um kleinere aber prestigeträchtige Anforderungen mit sogenannten „Begeisterungsfaktoren“ umzusetzen.



die zugehörige Linieneinheit das im Unternehmen standardmäßig eingesetzte Spezifikationstool. In solchen Fällen ist es besonders wichtig, beide Spezifikationen zueinander konsistent zu halten. Nur auf diese Weise lässt sich sicherstellen, dass das restliche nicht agil arbeitende Personal keine zusätzliche Einarbeitung braucht, um an alle für seine Arbeit erforderlichen Informationen zu gelangen.

Bei Medien- und/oder Tool-Brüchen, die bei derartigen Konstellationen nahezu immer auftreten, sollten die Artefakte untereinander durchgängig referenziert werden. Dadurch werden eine konsistente Zuordnung und Auffindbarkeit sichergestellt. Zum Beispiel kann im User Story Mapping Tool ein Link zum betreffenden Dokument oder auch Modell der Stan-



UM ZUM ERFOLG EINES HYBRIDEN PROJEKTMODELLS BEIZUTRAGEN, SOLLTE DER REQUIREMENTS ENGINEER ÜBER EIN TIEFERGREIFENDES TECHNISCHES VERSTÄNDNIS VERFÜGEN UND AUCH DIE ZUVOR BESCHRIEBENEN REGELN FÜR DIE AGILE SPEZIFIKATION BEFOLGEN.

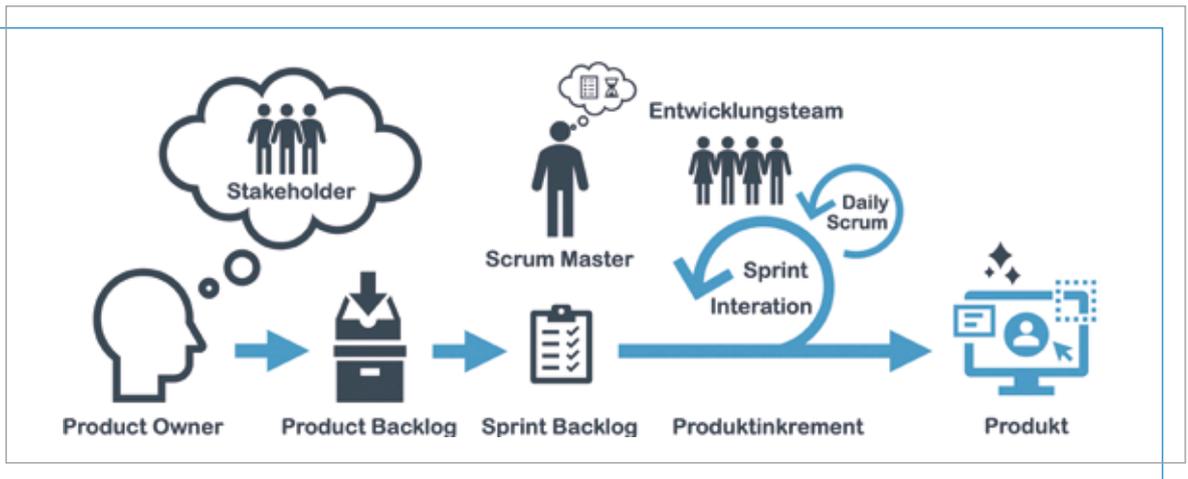
Alexa Ziesch, Senior Consultant,
DYNACON GmbH, www.dynacon.de

dard-Spezifikation ergänzt und im Gegenzug auch im betreffenden Dokument oder Modell die Nummer der zugehörigen User Story angegeben werden. So ist die Zuordnung zwischen den beiden Spezifikationen immer eindeutig.

Kommt für das agile Projekt ein User Story Mapping Tool zum Einsatz, empfiehlt sich außerdem das Benutzen von Tags zur Organisation der Stories. So können Tags der Status-Indizierung der User Stories dienen, zum Beispiel „Sprint OK“ wenn die User Story für einen Sprint ausreichend definiert ist und eingeplant werden kann, oder „Spec OK“ wenn die Spezifikation aus dem agilen Projekt im Standard-Spezifikationstool des Unternehmens nachgezogen wurde. Zur qualitativen und quantitativen Übersicht kann in den meisten Tools ein Report zu allen



Bild 1:
Vereinfachte
Darstellung des
Scrum-Flows



User Stories inklusive der beschriebenen Tags erzeugt werden.

User Stories vorbereiten

Um Sprints planen zu können, müssen die User Stories im Product Backlog vorbereitet werden (siehe auch Bild 1). Diese Tätigkeit übernehmen Requirements Engineers entweder in der Rolle des Product Owners (PO) oder bei größeren Projekten direkt als Teil des Scrum Teams. Das bedeutet nicht nur, dass die Anforderungen in Form von User Stories zu formulieren sind (Sprachmustervorlagen vereinfachen diese Aufgabe), sondern diese müssen auch priorisiert werden. Mit der Priorisierung wird auch die Reihenfolge für die nachher anstehende Entwicklung festgelegt. Aus diesem Grund ist es sinnvoll, Abhängigkeiten der Anforderungen untereinander, die Einfluss auf die Reihenfolge der Umsetzung der Anforderungen haben, direkt mit dem Scrum Team zu besprechen. So profitiert der Requirements Engineer von den Skills und Erfahrungen des Scrum Teams, das durch die Team-weite Einigung auf eine bestimmte Reihenfolge dem Product Owner einen Teil seiner Verantwortung abnimmt.

Doch mit welchen User Stories sollte begonnen werden? Priorisiert werden sollten vor allem diejenigen Anforderungen, die gleich zu Beginn den größten Wert für den späteren Benutzer schaffen, aber auch solche, die die Basis für Folgeaktivitäten oder auch Aktivitäten außerhalb des agilen Projekts darstellen. Sobald eine Abteilung

außerhalb des agilen Teams betroffen ist, empfiehlt es sich, zuerst die entsprechenden User Stories der übergreifenden Schnittstellen zu beschreiben. Durch die priorisierte Formulierung dieser Funktionen wissen die Betroffenen, welche Anforderungen sie in ihrer Domäne umsetzen müssen und können rechtzeitig einschätzen, welche Aufwände auf sie zukommen und wann diese einzuplanen sind.

In der agilen Welt gibt es einfache Schätzmethode, die den Product Owner bei seinen Planungsaufgaben unterstützen. Beispielsweise kann einer Anforderung ein bestimmter Wert zugeordnet werden (zum Beispiel in Form von Story Points). Alle anderen Funktionen, die danach entwickelt werden, werden dann an dieser Beispiel-Zuordnung gemessen. Je wertvoller eine Funktion für den Auftraggeber ist, desto mehr Story Points bekommt die zugehörige Anforderung. Nutzen kann der PO diese Zuordnung, indem er eben jene User Stories priorisiert, die durch verhältnismäßig wenig Aufwand einen hohen Nutzen (= viele Story Points) bringen. User Stories mit einem geringeren Wert und dafür erhöhtem Aufwand können zu einem späteren Zeitpunkt umgesetzt werden.

Klare Vorgaben für Spezifikation und Nachdokumentation

Die Unternehmensleitung sollte daher noch vor dem Start von agilen Projekten hinsichtlich des Spezifikationswerkzeugs eine klare Linie verfolgen. Die Empfeh-

lung: Die Erhaltung des Standard-Tools bis eine anderweitige Entscheidung getroffen wird (etwa die komplette Ablösung durch ein User Story Mapping Tool, wenn zukünftig ausschließlich agil entwickelt werden soll).

Pilotprojekte auch nachträglich im Standard-Tool zu dokumentieren, sollte nicht als doppelte Arbeit, sondern eher als eine vorausschauende Investition betrachtet werden. Hierdurch muss nicht das gesamte Personal des Unternehmens Schulungen durchlaufen, um auch noch im Nachhinein die Spezifikation lesen zu können. Außerdem sollte es unbedingt vermieden werden, nach Projektende zwei verschiedene Spezifikationsquellen zu haben. Sonst steht zu befürchten, dass die ehemaligen Projektbeteiligten immer wieder erklären müssen, wo welche Spezifikation anfängt und aufhört bzw. wie die Spezifikationen miteinander in Verbindung stehen. Oder es besteht das Risiko, Widersprüche zwischen den Spezifikationen festzustellen, weil die Spezifikation über Tool- und Medienbrüche hinweg nur schwer konsistent gehalten werden kann.

Übrigens: Erfolgt die Nachdokumentation im Standardtool erst nach Ende des agilen Projekts, hat das mehrere Vorteile. Durch die priorisierte Spezifikation im User Story Mapping Tool kann sichergestellt werden, dass das Scrum Team jederzeit arbeiten kann und nicht stillsteht (was zusätzliche Kosten verursachen würde).

Außerdem können alle Änderungen, die sich noch während des Projekts ergeben haben, direkt in die Spezifikation im Standardtool einfließen. Dadurch werden nachfolgende Korrekturläufe überflüssig. Es kann jedoch auch sein, dass dem Requirements Engineer nach dem Projekt keine Zeit mehr für diese Tätigkeit eingeräumt wird. In solchen Fällen empfiehlt es sich, die Spezifikationen im User Story Mapping Tool und im Standard-Spezifikationstool direkt parallel zu erstellen.

Wie sich die Arbeit eines Requirements Engineers in agilen Projekten ändert

Wird ein Produkt nach einem klassischen Vorgehensmodell entwickelt, ist es für den Requirements Engineer, der die zugehörige Spezifikation erstellt, in der Regel irrelevant, welcher Teil des Produkts zuerst entwickelt wird. Hierin unterscheidet sich die agile Vorgehensweise deutlich. Anstatt umfangreiche Konzepte, Lasten- oder Pflichtenhefte in großen Textdokumenten festzuhalten, muss ein agiles Projekt in viele kleine Häppchen aufgeteilt und mit einer User Story Mapping-Software verwaltet werden. Die Anforderungen werden vom Requirements Engineer (oftmals in der Rolle des Product Owners) in Form von User Stories im Product Backlog hinterlegt und priorisiert. Eine solche Priorisierung setzt voraus, dass sich der PO mit der späteren technischen Umsetzung seiner Anforderungen auseinandersetzt. Anderenfalls könnte er die Spezifikation nicht in sinnvolle Blöcke aufteilen und

Agile Vorgehensweisen sind vor allem bekannt durch ihre zahlreichen Review-Meetings und Retrospektiven

auch nicht beurteilen, welcher dieser Blöcke zuerst umgesetzt werden muss. Eine tiefere technische Einarbeitung als bei klassischen Entwicklungsprojekten ist sowohl bei einer hybriden als auch einer agilen Vorgehensweise für den Requirements Engineer unumgänglich.

Aber auch in der Kommunikation mit den Projektbeteiligten gibt es wesentliche Änderungen. Agile Vorgehensweisen sind vor allem bekannt durch ihre zahlreichen Review-Meetings und Retrospektiven (siehe Bild 2) – diese gibt es in klassischen Projekten meist nicht. Diese Meetings machen die Scrum-Methode besonders effektiv und helfen dabei, dass sich das Team schnell weiterentwickelt und in zukünftigen Sprints zügiger vorankommt. Indem die eigenen Prozesse reflektiert und kontinuierlich verbessert werden, können genauere Schätzungen abgegeben werden und das Team spielt sich besser aufeinander ein. Das agile Entwicklungsteam sorgt in den Sprint Reviews dafür, den Kunden von Anfang an in die Entwicklung einzubinden und ihm die Gelegenheit zu geben, Feedback zu

(Zwischen-) Ergebnissen zurückzumelden noch bevor das nächste Feature gebaut wird. Die typische Aufgabe eines Requirements Engineers, unter den Stakeholdern einen Konsens zu schaffen, wird an dieser Stelle auf alle Schultern, die an den Sprint Reviews teilnehmen, verteilt. Außerdem ermöglichen die Reviews dem Projektteam die unmittelbare Reaktion auf sich ändernde Anforderungen des Kunden. Dessen Rückmeldungen können durch den PO nach den Meetings einfach in Form von User Stories erfasst und in einen der nächsten Sprints eingeplant werden, sobald das nötige Refinement durchlaufen wurde.

Der Aufwand lohnt sich

Es lässt sich schlussfolgern, dass agile Projekte in klassischen Organisationsstrukturen mit Hilfe einer gut durchdachten Strategie erfolgreich durchgeführt werden können.

In Pilotprojekten mag dies zunächst einen erhöhten Zeitaufwand nach sich ziehen. Dieser erhöhte Aufwand ist jedoch nicht der agilen Vorgehensweise an sich, sondern vielmehr dem für diese Vorgehensweise notwendigen Change Management (also der Einführung der neuen Arbeitsweise) zuzuschreiben.

Hybrides Requirements Engineering kombiniert klassisches Anforderungsmanagement (dazu gehören die Erhebung, Analyse und die Priorisierung von Anforderungen) mit den Tätigkeiten eines Product Owners in iterativen Verfahren wie der Scrum-Methode. Um zum Erfolg eines solchen hybriden Projektmodells beizutragen, sollte der Requirements Engineer über ein tiefgreifendes technisches Verständnis verfügen und auch die zuvor beschriebenen Regeln für die agile Spezifikation befolgen. So wird am Ende eines Projekts ein Produkt entwickelt, das für Nutzer und/oder Kunden einen relevanten Mehrwert schafft und gleichzeitig durch konsistente Spezifikation eine lückenlose Dokumentation des Produkts sicherstellt.

Alexa Ziesch

